

CLAIMS:

1. A method for processing a set of primitive commands, the method comprising:
receiving one or more primitive commands, each primitive command including
information for processing vertex data using a user-developed program or
subroutine;
determining a set of related primitive commands from the one or more primitive
commands;
identifying a first primitive command for processing, the first primitive command
being a primitive command in the set of related primitive commands; and
transmitting a first program command to a processing engine for processing, the
first program command corresponding to the first primitive command.
2. The method of claim 1, further comprising the step of transmitting the one or more
primitive commands to a next fixed function engine if none of the one or more primitive
commands is a related primitive command.
3. The method of claim 1, further comprising the steps of determining whether to
transmit the first program command to the processing engine again for processing and
transmitting the first program command to the processing engine again for processing if
determining to do so.
4. The method of claim-1, further comprising the step of determining whether every

primitive command in the set of related primitive commands has been processed.

5

5. The method of claim 4, further comprising the step of transmitting the one or more received primitive commands that have not been processed to a next fixed function engine if every primitive command in the set of related primitive commands has been processed.

6. The method of claim 4, further comprising the steps of identifying a second primitive command for processing if every primitive command in the set of related commands has not been processed, the second primitive command being an unprocessed primitive command in the set of related primitive commands, and transmitting a second program command to the processing engine for processing, the second program command corresponding to the second related primitive command.

7. The method of claim 1, wherein the step of determining a set of related primitive commands comprises assembling a primitive using index information contained in each related primitive command.

8. The method of claim 1, wherein the first program command includes a pointer to a first user-developed program or subroutine.

9. The method of claim 8, wherein the first user-developed program or subroutine relates to primitive subdivision.

10. The method of claim 8, wherein the first user-developed program or subroutine relates to tessellation.
11. The method of claim 8, wherein the first user-developed program or subroutine relates to generating a vertex stream.
12. A method for processing a program command, the method comprising:
 - receiving a program command from a fixed function engine, the program command including information for processing vertex data using a user-developed program or subroutine;
 - retrieving the user-developed program or subroutine using a pointer to the user-developed program or subroutine included in the program command;
 - retrieving source data using a pointer to the source data included in the program command; and
 - performing specific operations on the source data based on instructions contained in the user-developed program or subroutine to produce processed vertex data.
13. The method of claim 12, further comprising the step of transmitting the processed vertex data to a memory buffer.
14. The method of claim 12, further comprising the step of transmitting results to the fixed function engine.

15. A fixed function engine configured to process a set of primitive commands, the fixed function engine comprising:

means for receiving one or more primitive commands, each primitive command including information for processing vertex data using a user-developed program or subroutine;

means for determining a set of related primitive commands from the one or more primitive commands;

means for identifying a first primitive command for processing, the first primitive command being a primitive command in the set of related primitive commands; and

means for transmitting a first program command to a processing engine for processing, the first program command corresponding to the first primitive command.

16. The fixed function engine of claim 15, wherein the first program command includes a program pointer and a data pointer.

17. The fixed function engine of claim 16, wherein the fixed function engine resides in a programmable primitive engine.

18. The fixed function engine of claim 17, wherein the programmable primitive engine is coupled to the processing engine, the processing engine configured to receive the first

program command, to retrieve a first user-developed program or subroutine using the program pointer and to retrieve vertex data using the data pointer, and to process the vertex data based on instructions included in the first user-developed program or subroutine, producing processed vertex data.

19. The fixed function engine of claim 18, wherein the processing engine is coupled to a storage resource, the storage resource configured to store the vertex data and the processed vertex data.

20. The fixed function engine of claim 18, wherein the first user-developed program or subroutine relates to primitive subdivision.

21. The fixed function engine of claim 18, wherein the first user-developed program or subroutine relates to tessellation.

22. The fixed function engine of claim 18, wherein the first user-developed program or subroutine relates to generating a vertex stream.

23. A processing engine configured to process a program command, the processing engine comprising:

means for receiving a program command from a fixed function engine, the program command including information for processing vertex data using a user-developed program or subroutine;

means for retrieving the user-developed program or subroutine using a pointer to the user-developed program or subroutine included in the program command;

means for retrieving source data using a pointer to the source data included in the program command; and

means for performing specific operations on the source data based on instructions contained in the user-developed program or subroutine to produce processed vertex data.

24. The processing engine of claim 23, further comprising means for transmitting the processed vertex data to a memory buffer.

25. The processing engine of claim 23, further comprising means for transmitting results to the fixed function engine.